METHOD OF PROVIDING TIME STAMPING SERVICE FOR SETTING CLIENT'S SYSTEM CLOCK

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates in general to an information security field, and more particularly to a method of providing a time stamping service for setting a client's system clock, wherein a service provider providing a public key infrastructure-based security service safely receives reference time information from a reliable third-party system and re-sets the client's system clock on the basis of the received reference time information, so as to assure the reliability of the client's system clock.

Description of the Prior Art

Recently, communication companies such as Korea Telecom have provided a public key infrastructure (PKI)-based security service together with an electronic data interchange (EDI) service for national pensions, electronic prescriptions, etc.

However, the PKI-based security service requires accuracy of time for verification of a certificate, but may not be normally provided due to time errors of a client, resulting in the leakage of incomings.

The main object of a typical time stamping service is to certify that a specific document has existed at a predetermined point of time and thus guarantee the accuracy of time.

For this reason, related documents have not referred to mechanisms for applying the time stamping service to the setting of a system clock of a client.

For the validity verification of a certificate using a certificate revocation list in connection with the public key infrastructure-based security service, there is no conclusion defined for a source of local time information as a benchmark for the validity verification and how to download the local time information from the source.

As a result, a system clock of a client employing the security service is generally used as the local time information. Provided that the system clock of the client is inaccurate, the security service will not be provided in spite of the fact that the certificate revocation list and certificate are valid.

SUMMARY OF THE INVENTION

Therefore, the present invention has been made in view of the above problems, and it is an object of the present invention to provide a method of providing a time stamping service for setting a client's system clock, which is capable of adding/defining new services to

predefined time stamp specifications for the setting of the client's system clock, modifying structures of TimeStampReq and TimeStampResp messages according to the service addition/definition, receiving time information from an objectively reliable time stamp using the modified message structures, and re-setting the client's system clock on the basis of the received time information.

In accordance with the present invention, the above and other objects can be accomplished by the provision of a method of providing a time stamping service for setting a client's system clock, comprising the first step of requesting the time stamping service of a time stamp authority server by a service requester; the second step of receiving the time stamping service request from the requester and creating and sending a response message corresponding thereto by the time stamp authority server; the third step of receiving the response message sent from the time stamp authority server and verifying the integrity thereof by the requester; the fourth step of downloading a certificate revocation list from a directory server and verifying the validity thereof by the requester; and the fifth step of downloading a certificate for an electronic signature of the time stamp authority server from the directory server, verifying an electronic signature value thereof and setting the client's system clock in accordance with the verified result by the requester.

BRIEF DESCRIPTION OF THE DRAWINGS

The above and other objects, features and advantages of the present invention will be more clearly understood from the following detailed description taken in conjunction with the accompanying drawings, in which:

Fig. 1 is a block diagram showing a hardware architecture for execution of a method of providing a time stamping service for setting a client's system clock in accordance with the present invention;

Figs. 2a to 2c are flowcharts illustrating the method of providing the time stamping service for setting the client's system clock in accordance with the present invention; and

Fig. 3 is a flowchart illustrating a procedure of verifying the validity of a certificate revocation list in accordance with the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

With reference to Fig. 1, there is shown in block form a hardware architecture for execution of a method of providing a time stamping service for setting a client's system clock in accordance with the present invention. In this drawing, the reference numerals 1 and 2 denote first and second clients requesting time information, respectively. The first client 1 has a personal computer (PC) environment, and the second client 2 has a

UNIX environment. A time stamp authority (TSA) server 3 is operable in a UNIX-based system to provide reliable time information. A directory server 4 is adapted to manage certificates for electronic signatures of the TSA server 3. This server 4 is one of unit systems for constructing a public key infrastructure, and manages certificates of all objects and a certificate revocation list. The Internet 5 is a fundamental communication network among the first and second clients 1 and 2, TSA server 3 and directory server 4, and is based on a transmission control protocol/Internet protocol (TCP/IP) network or a telephone accessing network such as a point-to-point protocol (PPP) network.

A description will hereinafter be given of the operation of the present invention under the above-stated hardware architecture with reference to Figs. 2a to 2c.

Figs. 2a to 2c are flowcharts illustrating the method of providing the time stamping service for setting the client's system clock in accordance with the present invention. This method basically comprises five steps.

Namely, the present method comprises the first step of requesting the time stamping service of the TSA server 3 by a service requester, the second step of receiving the time stamping service request from the requester and creating and sending a response message corresponding thereto by the TSA server 3, the third step of receiving the response message sent from the TSA server 3 and verifying the integrity thereof by the

requester, the fourth step of downloading a certificate revocation list from the directory server 4 and verifying the validity thereof by the requester, and the fifth step of downloading a certificate for an electronic signature of the TSA server 3 from the directory server 4, verifying an electronic signature value thereof and setting a system clock of the client 1 or 2 in accordance with the verified result by the requester.

The first step is composed of steps S21 to S23 in Fig. 2a.

At the first step, the requester first generates a random number with a given value and sets it as a nonce value of a service request message, or TimeStampReq message (S21).

In order to inform the time stamp authority server that the service request is for the setting of the client's system clock, the requester sets a requestType parameter of the TimeStampReq message, defined in the present invention, to 2, or a getBaseTime value, and adds the resulting structure to an extension field of the TimeStampReq message (S22).

Subsequently, the requester fills other parameters of the TimeStampReq message with given values and then sends the resulting TimeStampReq message to the TSA server 3 (S23).

The second step is composed of S24 to S28 in Fig. 2a.

Namely, the second step includes a sequence of steps processed by the TSA server 3. At the second step, the TSA server 3 first receives a service request message, or TimeStampReq message, sent from the requester (S24).

Then, the TSA server 3 authenticates and verifies the received TimeStampReq message (S25).

If there is an error at the above step S25, the TSA server 3 processes the received TimeStampReq message as an erroneous message, sends the processed result to the requester and ends the corresponding process.

However, if there is no error at the above step S25, the TSA server 3 fills parameters of the response message, or TimeStampResp message, with given values (S26).

In order to assure the integrity of the response message, the TSA server 3 extracts a TSTInfo structure from a TimeStampResp message structure created at the above step S26 and, in turn, current time information, or a genTime value, from the extracted TSTInfo structure, calculates a message authentication code (MAC) value on the basis of the extracted genTime value and a nonce value, set by the requester and contained in the TimeStampReq message, and then sets the calculated MAC value and identifier information of an algorithm used for

the calculation of the MAC value respectively in corresponding fields of a MacInfo structure proposed in the present invention (S27).

Subsequently, the TSA server 3 adds the resulting MacInfo structure to an extension field of the TSTInfo structure and thus completes the creation of the TimeStampResp message structure proposed in the present invention.

Thereafter, the TSA server 3 sends the response message, or TimeStampResp message, created through the above steps to the requester (S28).

The third step is composed of S29 to S34 in Fig. 2b.

At the third step, the requester first receives the response message, or TimeStampResp message, sent from the TSA server 3 (S29) and authenticates and verifies the received response message (S30).

If there is an error such as an ASN.1 NOTATION error at the above step S30, the requester processes the received response message as an erroneous message.

If there is no error at the above step S30, the lower-order steps beginning with step S31 are performed. That is, the requester extracts a TSTInfo structure from the TimeStampResp message and directly calculates a MAC

value to check the integrity of the TimeStampResp message (S31).

For the calculation of the MAC value at the above step S31, the requester first extracts current time information, or a genTime value, from the extracted TSTInfo structure and finds a nonce value, set by the requester and sent to the time stamp authority server.

The requester directly calculates a MAC value on the basis of the extracted genTime value and the found nonce value.

Then, the requester verifies the calculated MAC value to check whether the integrity of the received response message has been assured.

For the integrity verification, the requester first extracts a MacInfo structure proposed in the present invention from the TimeStampResp message sent from the time stamp authority server and, in turn, a MAC value from the extracted MacInfo structure and then compares the extracted MAC value with the MAC value calculated at the above step S31 to determine whether the two MAC values are equal (S32).

If the two MAC values are not equal at the above step S32, the requester recognizes that the current time information, or the genTime value, sent from the TSA server 3 was altered during the sending (S33) and the system clock of the client 1 or 2 cannot thus be set

because the integrity of the received response message has not been assured. As a result, the requester processes the received response message as an erroneous message (S34).

To the contrary, in the case where the two MAC values are equal at the above step S32, the requester recognizes that the integrity of the received response message has been assured and thus performs the following fourth step.

The fourth step is composed of S35 to S37 in Fig. 2b.

At the fourth step, the requester first downloads the certificate revocation list (CRL) and the certificate for the electronic signature of the TSA server 3 from the directory server 4, which manages certificates of all objects and the certificate revocation list (S35).

In order to verify the validity of the certificate revocation list downloaded from the directory server 4 on the basis of a genTime value contained in the response message sent from the TSA server 3, the requester extracts time information set to thisUpdate and nextUpdate values from the certificate revocation list (S36).

Then, the requester determines whether the genTime value is present between the thisUpdate and nextUpdate values, so as to determine whether the certificate revocation list is valid (S37).

Upon determining at the above step S37 that the certificate revocation list is not valid, the requester recognizes that a signature value sent from the TSA server 3 (contained in a signature value field of a SignerInfo structure) cannot be verified (S38) and the system clock of the client 1 or 2 cannot thus be set. As a result, the requester performs an associated error process (S39).

However, if the CRL is valid at the above step S37, the requester proceeds to the fifth step.

The fifth step is composed of steps S40-S51 in Fig. 2c.

At the fifth step, the requester finally determines whether a genTime value sent from the TSA server 3 is reliable, by verifying a signature value sent from the TSA server 3.

First, in order to verify the validity of the certificate for the electronic signature of the TSA server 3, the requester extracts desired information (S40) from the certificate of the TSA server 3 and checks whether a serial number of the certificate of the TSA server 3 among the extracted information is present in the certificate revocation list (S41).

In the case where the serial number of the certificate of the TSA server 3 is present in the certificate revocation list at the above step S41, the

requester recognizes that the signature value sent from the TSA server 3 cannot be verified and the system clock of the client 1 or 2 cannot thus be set, and then performs an associated error process (S42 and S43).

To the contrary, if the serial number of the certificate of the TSA server 3 is not present in the certificate revocation list at the above step S41, the requester performs a pre-process for the verification of the signature value sent from the TSA server 3.

Namely, the requester extracts a public key from the certificate for the electronic signature of the TSA server 3, downloaded from the directory server 4.

Then, the requester extracts the signature value from a SignerInfo structure of the TimeStampResp message, decodes the extracted signature value using the extracted public key and extracts a hash value (referred to hereinafter as M1), or a digest value, from the decoded result (S44).

Thereafter, the requester directly calculates a hash value (referred to hereinafter as M2) using a digest algorithm of the SignerInfo structure (S45).

Subsequently, the requester compares the two hash values, or M1 and M2, with each other to determine whether they are equal (S46). If M1 and M2 are not equal, the requester recognizes that the time stamp authority server sending the TimeStampResp message is not valid and the

client's system clock cannot thus be set, and then performs an associated error process (S47 and S48).

However, if M1 = M2, the requester recognizes that the TSA server 3 sending the TimeStampResp message is valid (S49).

Then, the requester sets the client's system clock on the basis of a genTime value extracted from the TimeStampResp message (S50) and then performs the subsequent service (S51).

Fig. 3 is a flowchart illustrating a procedure of verifying the validity of a certificate revocation list in accordance with the present invention. Through a sequence of steps in Fig. 3, the PKI-based security service cannot be provided when the client's system clock is not accurately set. First, the requester downloads the CRL from the directory server 4 and decodes it (S1 and S2).

Then, the requester extracts available time information of the CRL from the CRL and current time information, or a Tcurrent value, from the client, respectively (S3 and S4).

Thereafter, the requester determines whether the Tcurrent value is present between thisUpdate and nextUpdate values, namely, thisUpdate < Tcurrent < nextUpdate (S5). If the Tcurrent value is not present between the thisUpdate and nextUpdate values, the requester recognizes that the verification of the

certificate validity ends in failure (S6); otherwise, it extracts a revoked certificates structure from the CRL (S7). Then, the requester determines whether a desired certificate is present in the extracted revoked certificates structure (S8). If the desired certificate is not present in the extracted revoked certificates structure, the requester recognizes that it was abrogated (S9). However, in the case where the desired certificate is present in the extracted revoked certificates structure, the requester recognizes that it is valid (S10).

As apparent from the above description, the present invention provides a method which is effectively connected with a nonrepudiation service to objectively certify that a specific document has existed at a predetermined point of time. The present method provides a time stamping service for providing objectively reliable standard time information to a requester so that a client's system clock can be set on the basis of the standard time information. Therefore, the reliability and objectivity of the client's system clock can be assured.

Further, the present method is effectively connected with a public key infrastructure-based security service to overcome security service obstacles resulting from an inaccurate system clock of a client.

Although the preferred embodiments of the present invention have been disclosed for illustrative purposes, those skilled in the art will appreciate that various

modifications, additions and substitutions are possible, without departing from the scope and spirit of the invention as disclosed in the accompanying claims.